

IN THE CLAIMS:

Please cancel claim 14 without prejudice.

- 1 1. (Currently Amended) A system comprising:
2 a plurality of network resources adapted to process received block-based protocol
3 data access requests; and
4 a plurality of ~~one or more~~ virtual servers each comprising a logical partitioning of
5 the network resources to establish an instance of a multi-protocol server configured to
6 service the block-based data access requests by converting the block-based protocol
7 requests to appropriate file system data requests, each virtual server allowed shared
8 access to resources of the file system; and
9 a context data structure provided to each virtual server, the context data structure
10 including information pertaining to a security domain of the virtual server for each
11 supported access protocol, to enable controlled access to the shared resources of the file
12 system.
- 1 2. (Original) The system of claim 1 wherein the network resources comprise
2 network interfaces assigned to one or more network address resources.
- 1 3. (Previously Presented) The system of claim 1 further comprising storage media
2 configured to store information as units of storage resources, the units of storage
3 resources allocated among each of the virtual servers.
- 1 4. (Original) The system of claim 3 wherein the units of storage resources comprise
2 volumes.
- 1 5. (Original) The system of claim 3 wherein the units of storage resources comprise
2 qtrees.

1 6. (Previously Presented) The system of claim 3 further comprising an operating
2 system having a file system resource adapted to perform a boundary check to verify that a
3 request is allowed to access certain units of the storage resources on the storage media,
4 each virtual server allowed shared access to the file system and further adapted to create
5 virtual disks within the units of storage resources and wherein each of the virtual disks
6 associated with one or more of the virtual servers.

1 7. (Previously Presented) The system of claim 6 wherein the operating system
2 further comprises a user interface having a command set adapted to operate on virtual
3 disks, and wherein the command set executes within a context of a virtual server.

1 8. (Original) The system of claim 7 wherein the user interfaces comprises a
2 command line interface (CLI) adapted to support the command set.

1 9. (Previously Presented) The system of claim 8 wherein the CLI comprises a lun
2 command adapted to perform operations to a virtual disk associated with the context of
3 the virtual server.

1 10. (Previously Presented) The system of claim 9 wherein the lun command creates a
2 logical unit number on a file system associated with the server, the logical unit number
3 being associated with the context of the virtual server.

1 11. (Original) The system of claim 8 wherein the CLI comprises an igroup command
2 that generates a set of file system primitive for binding an initiator group to one or more
3 initiator addresses and wherein the initiator group is associated with the context of the
4 virtual server.

1 12. (Original) The system of claim 1 wherein the block-based protocol comprises
2 iSCSI.

1 13. (Original) The system of claim 1 wherein the block-based protocol comprises
2 FCP.

1 14. (Cancelled).

1 15. (Original) The system of claim 1 wherein the multi-protocol server is further
2 adapted to process data access requests in response to one or more file-level protocols.

1 16. (Currently Amended) A method for implementing a virtual server, the method
2 comprising the steps of:
3 adapting a plurality of network resources to process received block-based protocol
4 data access requests; ~~and~~
5 partitioning the network resources to establish ~~one or more~~ a plurality of virtual
6 servers, each comprising an instance of a multi-protocol server configured to service the
7 block-based data access requests by converting the block-based protocol requests to
8 appropriate file system primitives; ~~and~~
9 providing a context data structure to each virtual server, the context data structure
10 including information pertaining to a security domain of the virtual server for each
11 supported access protocol, to enable controlled access to the shared resources of the file
12 system.

1 17. (Previously Presented) The method of claim 16 further comprising the step of
2 configuring storage media to store information as units of storage resources, the units of
3 storage resources allocated among each of the virtual servers.

1 18. (Original) The method of claim 17 wherein the units of storage resources
2 comprise volumes.

1 19. (Original) The method of claim 17 wherein the units of storage resources
2 comprises qtrees.

1 20. (Currently Amended) A computer readable medium containing executable
2 program instructions for implementing a virtual server, the executable program
3 instructions comprising program instructions for:
4 adapting a plurality of network resources to process received block-based protocol
5 data access requests; ~~and~~
6 partitioning the network resources to establish ~~one or more~~ a plurality of virtual
7 servers, each comprising an instance of a multi-protocol server configured to service the
8 block-based data access requests by converting the block-based protocol requests to
9 appropriate file system primitives; and
10 providing a context data structure to each virtual server, the context data structure
11 including information pertaining to a security domain of the virtual server for each
12 supported access protocol, to enable controlled access to the shared resources of the file
13 system.

1 21-23. (Cancelled).

1 24. (Currently Amended) A method, comprising:
2 receiving a block-based data access request from a client;
3 forwarding the request to a virtual server;
4 performing security checks on the request using a context data structure provided
5 to each virtual server, the context data structure including information pertaining to a
6 security domain of the virtual server for each supported access protocol, to enable
7 controlled access to the shared resources of the file system;
8 converting the received block-based data access request to a file system data
9 access request;
10 in the event that the request passes the security checks, servicing the file system
11 data access request to generate a response; and
12 forwarding the generated response to the client.

1 25. (Currently Amended) A system, comprising:
2 a network interface to receive a block-based data access request from a client;
3 the network interface to forward the request to a virtual server;
4 the operating system to perform security checks on the request using a context
5 data structure provided to each virtual server, the context data structure including
6 information pertaining to a security domain of the virtual server for each supported
7 access protocol, to enable controlled access to the shared resources of the file system;
8 in the event that the request passes the security checks, a process to convert the
9 received block-based data access request to a file system data access request;
10 the process to service the file system data access request to generate a response;
11 and
12 the process to forward the generated response to the client.

1 26. (Currently Amended) A computer readable media, comprising:
2 said computer readable media containing instructions for execution on a
3 processor for the practice of a method, the method comprising:
4 receiving a block-based data access request from a client;
5 forwarding the request to a virtual server;
6 performing security checks on the request using a context data structure
7 provided to each virtual server, the context data structure including information
8 pertaining to a security domain of the virtual server for each supported access
9 protocol, to enable controlled access to the shared resources of the file system;
10 converting the received block-based data access request to a file system data
11 access request;
12 in the event that the request passes the security checks, servicing the file
13 system data access request to generate a response; and
14 forwarding the generated response to the client.